

WEST**Freeform Search****Database:**

US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term:**Display:** **Documents in Display Format:** CIT Starting with Number **Generate:** Hit List Hit Count Image **Search History****Today's Date:** 9/12/2001

<u>DB Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	l22 and create same folder with documents	104	<u>L24</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	l22 and create same folder without documents	3852126	<u>L23</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	l21 and create same folder	413	<u>L22</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	document and folder or workfolder	3247	<u>L21</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/204)!.CCLS.))	381	<u>L20</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/511)!.CCLS.))	116	<u>L19</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/\$)!.CCLS.))	11438	<u>L18</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((701/\$)!.CCLS.))	18935	<u>L17</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((704/\$)!.CCLS.))	11520	<u>L16</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((717/\$)!.CCLS.))	2940	<u>L15</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((717/3)!.CCLS.))	235	<u>L14</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((704/9)!.CCLS.))	469	<u>L13</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((701/30)!.CCLS.))	73	<u>L12</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((701/29)!.CCLS.))	592	<u>L11</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/531)!.CCLS.))	665	<u>L10</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/522)!.CCLS.))	67	<u>L9</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/500)!.CCLS.))	276	<u>L8</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/206)!.CCLS.))	217	<u>L7</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/200)!.CCLS.))	687	<u>L6</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/104.1)!.CCLS.))	1175	<u>L5</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/104)!.CCLS.))	3	<u>L4</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/100)!.CCLS.))	820	<u>L3</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/10)!.CCLS.))	1492	<u>L2</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	((707/1)!.CCLS.))	1149	<u>L1</u>

WEST[Help](#)[Logout](#)[Interrupt](#)[Main Menu](#) [Search Form](#) [Posting Counts](#) [Show S Numbers](#) [Edit S Numbers](#) [Preferences](#)**Search Results -**[Terms](#)[Documents](#)

14 and create with folder

198

Database:

US Patents Full-Text Database
US Pre-Grant Publication Full-Text Database
JPO Abstracts Database
EPO Abstracts Database
Derwent World Patents Index

IBM Technical Disclosure Bulletins

[Refine Search:](#)[Clear](#)**Search History****Today's Date:** 9/12/2001

<u>DB Name</u>	<u>Query</u>	<u>Hit Count</u>	<u>Set Name</u>
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	14 and create with folder	198	L7
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	15 and folder with creation	55	L6
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	12 and (exist\$ or non-exist\$)	54064	L5
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	12 and exist\$ or non-exist\$	63067	L4
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	12 and folder with creation	55	L3
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	11 and exist\$	54064	L2
USPT,PGPB,JPAB,EPAB,DWPI,TDBD	documents	214007	L1

Generate Collection

L24: Entry 24 of 104

File: USPT

May 4, 1999

DOCUMENT-IDENTIFIER: US 5899995 A

TITLE: Method and apparatus for automatically organizing information

ABPL:

The present invention discloses an electronic filing system for automatically organizing information into a number of storage elements or folders. The storage elements are folders arranged hierarchically. The information is stored into the storage elements based on its contents. Each storage element is associated by a set of profiles and a set of constraint expressions. The constraint expressions are quantitative measurements of the profiles. The information is scanned and its features are extracted and compared with the profiles. If the features match the storage element's profiles, the features are quantitatively evaluated to produce information expressions. If the information expressions are within the constraints specified by the constraint expressions, the information is stored into the storage element. The profiles and the constraint expressions can be adjusted to reflect the changes in the user's interests.

BSPR:

The present invention relates to the field of automatic organization of information. In particular, the invention relates to smart folders for storing files based on their contents and profiles of usage.

BSPR:

There are a number of ways to help organize and store electronic information. One way is to increase the amount of mass storage capacity by upgrading the hard drive. It is not uncommon to find home machines with hard drives having storage capacities greater than 1 Gigabyte (GB). However, increasing storage capacity does not solve the problem of organizing data. Faced with a large amount of information, the user typically creates more directories or folders to organize and store the information for later retrieval. However, the user usually forgets the context under which he or she files the information, and will thus have to spend a large amount of time in retrieving the information.

BSPR:

Another way to improve the organization of files is the use of rules to govern the depositing of information in application specific folders. Examples of systems employing this technique are Eudora, CCMail, and several USENET news readers. Although this rule-based technique provides the user with some control over his or her organization of the files, it has many disadvantages. First, this rule-based disposition of messages or documents is very application-specific and applies only to messages or documents within the context of the specific application. For example, one cannot file a Word.TM. document in a Eudora folders without constructing an e-mail message. As another example, one must start CCMail to look at a PowerPoint presentation in a CCMail folder. Second, these rule-based systems are static and inflexible. Once a rule is set up, it does not change and cannot be adjusted unless the user modifies them specifically. Because of this, the rules can be very complex, and time consuming to maintain.

BSPR:

Another way to improve the organization of files is the use of indexing. One example of a system employing this technique is Wide Area Information Servers (WAIS). This method allows a user to index folders of documents by their contents. These systems allow the user to quickly search for documents with specific content. The user may also search for documents having a similar content with another document without specifying the details of the content. However, these systems do not provide automatic categorization and organization. The categorization function allows the user to file documents in multiple locations and allows a much more natural mode of information access.

BSPR:

It is therefore desirable to have an intelligent filing system that can automatically organize the information into folders based on the contents and profiles of usage.

DRPR:

' FIG. 3 is an illustrative screen display of one embodiment of the categorization of the folders.

DRPR:

FIG. 4A is an illustration of a screen display of one embodiment of the smart folder profile.

DRPR:

FIG. 4B is an illustration of a screen display of one embodiment of the smart folder constraints.

DRPR:

FIG. 5 is a structure diagram showing one embodiment of the processing of an input document from a parent smart folder to folders at lower levels.

DRPR:

FIG. 6 is a structure diagram showing one embodiment of the processing modules in the smart folder system.

DEPR:

The present invention discloses a method and apparatus for automatically organizing information such as documents in an electronic filing system based on contents and usage profiles.

DEPR:

The term "SmartFolder" is used to refer to any storage element that stores the information based on contents and/or usage profiles. The term "InfoEvaluator" refers to a system, apparatus, or a program, that is capable of evaluating the information and providing an information value representing the significance of the information with respect to some criteria. The term "Intelligent Filing System" (IFS) is used to refer to the system that automatically organizes information into storage elements based on contents and usage profiles. An example of such a system is any processor-based system such as a PC or a workstation. The "SmartFolder Intelligent Filing System" may be installed on the system as shown in FIG. 1. It is apparent to one of ordinary skill in the art that other microprocessor-based systems may implement the Smart Folder IFS.

DEPR:

The SmartFolder Intelligent Filing System (IFS) allows the user to define a set of rules that allow the document to be filed in multiple contexts of the user's choosing. This automatic filing process is less susceptible to lapses of human memory, and the rules can be defined in terms of "learning" profiles that automatically adjust as the user organizes the desktop. The documents are categorized by their contents. Therefore, the tedious process of searching based on contents is eliminated.

DEPR:

The SmartFolder IFS is document-oriented rather than application-oriented. The filing of the documents is independent of the application (e.g., Eudora, CCMail, USENET news readers) that is supplying the document. The SmartFolder IFS allows the user to set up a rule based on a learning "profile" which captures the history of usage of the user over time. The profile can adjust according to user's actions to become better and better at categorizing the user's documents, even if the user cannot state exactly what the rules are for the categorization. In other words, the SmartFolder IFS "learns" the rules based on how the user arranges the information.

DEPR:

Indexing is the traditional method for searching documents in an electronic filing system. Although the SmartFolder IFS uses indexing, the indexing is used merely for labeling purpose. The categorization function of the SmartFolder IFS allows the user to file documents in multiple locations and allows a much more natural mode of information access than specific searches based on indexing only.

DEPR:

In the SmartFolder IFS, all documents are stored in a unified and industry standard format. For example, the Microsoft Structured Storage facility of OLE2.0 provides this standard file format. This allows the SmartFolder IFS and the Viewer applications to interact without explicit knowledge of each other.

DEPR:

When a document in Structured Storage Format is submitted to a SmartFolder, the document is abstracted into a list of features or attributes. These features are then used by the SmartFolder Constraint system to evaluate the candidate document for this folder. The extracted features can be described in three categories:

DEPR:

The standard attributes of a file, its name, size, date of creation, etc. are features that can be used for SmartFolder. This allows rules like, "new Word documents larger than 1 Mb go in this folder".

DEPR:

The Structured storage file format defines a standard property set for a document that includes features such as subject/title, author, abstract, date/time originated, etc. These features can also be used for the SmartFolder IFS rules. In addition, the SmartFolder IFS will include recognized features, Uniform Resource Locators (URLs), Email addresses, names, place names, dates/times, and the like that can be extracted from the document text using pattern matching algorithms. And finally, each term in the document is also a feature, so traditional rules like "documents that contain the term 'Intel' go in this folder" can be specified by the user.

DEPR:

In addition to attributes that are directly extracted from the document, the SmartFolder IFS will use attributes that are derived from this document's relationship to the user, or to other documents in the folder. Some examples of the compound attributes are the evaluation value (Value) and the WAIS score. The 'Value' of a document from the InfoEvaluator system (scored against the user's profile) can be used just as any other feature of the document (see "Constraints and Profiles" below). The WAIS indexing system has the facility to be able to calculate a 'weight' for a document based on it's similarity to other documents. This weight value could be used to specify SmartFolder rules like "This folder accepts documents that are like the other documents that are already here".

DEPR:

The storage management of the SmartFolder IFS allows the construction of information categorizers by letting the user build hierarchies of folders. Each folder, when it receives a document, attempts to give that document to each of its child folders. So the higher level folders provide a pre-filter for their children. These hierarchies can be arbitrarily deep so the user is free to build as simple or as complex a categorization system as is necessary.

DEPR:

The addition of a standard "categorize" dialog allows the user to avoid the "where do I put this where I can find it again?" question when saving a new document. Rather than deciding what single context the document should be placed in, the user can assign multiple contexts, and retrieve the document from any one of those contexts. With the addition of long file names in newer PC operating systems, the user's ability retrieve documents by their context is greatly enhanced.

DEPR:

As mentioned above, a SmartFolder is associated with a set of constraints or rules that govern what documents are eligible to be placed in that folder. These rules can refer to indirect attributes that are calculated by evaluating the feature set of the document against a user profile. This facility is implemented using the InfoEvaluator technology one embodiment of which is described in the U.S. patent application, "Information Evaluation", Ser. No. 08/367,991 filed Jan. 3, 1995, now U.S. Pat. No. 5,799,304 assigned to the same assignee of the present invention. It is apparent to one of ordinary skill in the art that other information evaluation techniques as known in the art, may be utilized.

DEPR:

As the user uses the "Categorize" dialog or moves documents from a SmartFolder to another, the InfoEvaluator Profiles can be trained to more accurately reflect the user's desired organization. This facility allows the SmartFolder IFS to be adaptive to changes in the user's interests or organizational methods. The detailed description of the categorize dialog facility will be provided in the following.

DEPR:

FIG. 2A illustrates a screen display 100 on display monitor 62 showing one embodiment of a hierarchical arrangement of the various folders and files in accordance with the teachings of the Smart Folder System of the present invention. In one embodiment, the display format shown in screen display 100 is similar to that of Windows.RTM.95 which is an operating system developed by Microsoft Corporation of Redmond, Wash. However, the teachings of the present invention may be practiced without utilizing any specific operating system or computing platform.

DEPR:

Screen display 100 comprises Title bar 110, Menu bar 120, Open Folder 130, Display Window 140, and Scroll bar 190. Title bar 110 indicates the selected program and folder. On this screen display, the program SmartFolder is located at drive D in hard drive 56 in FIG. 1.

DEPR:

Open Folder 130 shows the name of the folder currently opened together with the icons in the tool bar. These icons are the standard icons in the Windows.RTM.95 platform. For example, Folder Up One Level 132 is used to go up to the parent folder. Assign Network Drive 134 is used to assign a drive letter to a network drive. Icons Cut 136 and Copy 138 are used for operations that move or copy files or folders.

DEPR:

Display window 140 shows one embodiment of the file organization architecture. The organization of the filing system is similar to a tree structure. Folders or documents are arranged at various hierarchical levels as branches in a tree. The Root Folder 145 is the highest selected folder in this filing system. As shown in FIG. 2A, the Root Folder Icon 145 is an open folder with the name SmartFolder. Under this root are the first level folders. For this illustrative example, there are four first-level folders: First-level folders 150, 160, 170, and 180 named Intel, New Media Projects, Projects, and Stock, respectively. Each of these first-level folders may in turn have one or more second-level folders. For this illustrative example, First-level folder Intel 150 has three second-level folders 152, 154, and 156, with names Advertisements, Group, and Presentations, respectively. First-level folder New Media Projects 160 has two second-level folders 162 and 164 with names Staff and Status Reports, respectively. First-level folder Projects 170 has two second-level folders InfoEvaluator 172 and SmartFolder 176. First-level folder Stock 180 has no second-level folders.

DEPR:

Second-level Folder InfoEvaluator 172 has a third-level folder Status Reports 174. Second-level folder SmartFolder 176 has a third-level folder Status Reports 178.

DEPR:

Referring to FIG. 2B, a block diagram illustrating one embodiment of the system architecture for the SmartFolder Intelligent Filing System (IFS) in the present invention is shown. The SmartFolder IFS 200 comprises a Reporter 210, a Feature Set 220, a Feature Recognizer 230, a Constraint Evaluator 250, a set of Folder Constraints/Profiles 240, a Categorize Dialog 260, and a Storage Manager 270.

DEPR:

The Reporter 210 scans the documents to pull out the relevant documents based on some preprogrammed key words. The Reporter 210 may be integrated to specific services (e.g., Email, USENET news, news wire service, on-line services) to files stories as they arrive, or to contact the service on a periodic basis to retrieve the relevant documents.

DEPR:

The Feature Recognizer 230 takes the results from the Reporter 210 and extracts and recognizes the features relevant to the documents in accordance to the selected folders. These features are forwarded to the Feature Set 220 and the Constraint Evaluator 250.

DEPR:

The Feature Set 220 contains all the features characterizing the documents. The Feature Set 220 may be augmented by the newly extracted features from the Feature Recognizer 230. These features are forwarded to the Constraint Evaluator 250 for evaluation.

DEPR:

The Constraint Evaluator 250 receives the features of the documents from the Feature Recognizer 230, the set of features from the Feature Set 220, and the constraints and profiles from the Folder Constraints/Profiles 240. The Constraint Evaluator 250 evaluates the features of the documents based on the profiles, and interacts with the Categorize Dialog 260 before forwarding the document to the Storage Manager 270.

DEPR:

The Categorize Dialog 260 provides an interactive session during which the user selects the folders that would accept the current documents. The description of the Categorize Dialog 260 is discussed later.

DEPR:

The Storage Manager 270 receives the document processed by the Constraint Evaluator 250 and optionally interacts with the user through the Categorize Dialog 260. The Categorize Dialog 260 allows the user to modify the folders or storage areas with the document to be filed. The Storage Manager 270 then files the document into appropriate folders and/or storage areas.

DEPR:

Referring to FIG. 3, a screen display of a session illustrating the operation of the Categorize Dialog process is shown. The Categorize Screen 300 shows an example of a

interactive session between the user and the Categorize Dialog process. Toolbar 310 shows the session name, in this case, Categorize. Selected Folders 320 shows the list of those folders selected by the user to contain the document being filed. Selected Folder List 330 shows the names of the folders selected to contain the document. Unselected Folders 340 shows the list of the folders that are not selected to contain the document. Unselected Folder List 350 shows the names of the folders that are not selected to contain the document. Left Arrow 360 moves the highlighted folder selected by the user from Unselected Folder List 350 to Selected Folder List 330. Right Arrow 370 moves the highlighted folder selected by the user from Selected Folder List 320 to Unselected Folder List 350.

DEPR:

Selection of the icon OK 380 ends the categorize session with the selected folders shown in the Selected Folder List 330. The icon Cancel 382 cancels the categorize session and restores the original Selected Folder List 320 and Unselected Folder List 350. The icon Help 384 provides a help menu to guide the user through the categorize session. The icons Network 386 and Down Arrow 388 provide access to Network.

DEPR:

Each folder has a list of constraints and profiles initially created by the user. A document is put in a particular folder if it satisfies the folder's constraints and fits its profiles. The profiles of a folder are the key words or features that characterize the contents of the documents stored in the folder. For example, a user may create the profiles of a folder including terms such as stock, Intel, Microsoft having equal significance. Therefore, a story about the rising of Microsoft's stocks after the delivery of a new network software package seems to fit the profiles of the folder. The constraints of a folder provide a quantitative means to characterize the folder's properties including the profiles and other characteristics. For example, the constraints of a folder may include an interesting factor as evaluated by the InfoEvaluator of 50 (from a range of 0 to 100) and a file date of less than 2 days.

DEPR:

FIGS. 4A and 4B show an example of a session to create the folder's profiles and constraints. Referring to FIG. 4A, the Folder Properties Screen 400 shows the folder properties for the profiles. Menu bar SmartFolder Properties 410 indicates that the properties of the folder are being entered. Properties Icon OK 412 ends the session using the entered properties. Properties Icon Cancel 414 ends the session without changing the folder's properties.

DEPR:

SmartFolder Properties 410 includes four tabs: General 420, Sharing 422, Profile 430, and Constraints 460. General 420 includes general items about the folder such as names, date when first created, size, etc. Sharing 422 shows the list of other folders that shares the documents contained in the folder. Profile 430 shows the profile of the folder. Constraints 460 show the constraints and the combinations of these constraints of the folder.

DEPR:

Profile List 435 contains the list of the features or key words that characterize the contents of the documents contained in the folder. Folder List 435 has two components: Name and Vote. Profile Name is the name of the profile or key word, and Vote is the voting power of the corresponding profile name. Select Name 440 displays the name of a key word to be added or deleted. Select Vote 442 shows the voting power for the corresponding key word in Select Name 440. New Agent 450 indicates if the Select Name 440 is added to the Profile List 435. Delete Agent 452 is used to delete or remove the Select Name 440 from the Profile List 435. Profile Icon OK 454 ends the profile selection. Profile Icon Cancel 456 cancels the profile selection and restore back to the original profile.

DEPR:

Referring to FIG. 4B, an example of the constraints of a folder is shown. Constraints 460 include Attribute 470, Operator 480 and Value 490. Attribute 470 contains a list of the folder attributes that can be used as constraints. Typical attributes are the value as evaluated by the InfoEvaluator and the file date. Operator 480 contain the corresponding operators that link the attributes to the values. Typical operators are greater than (>), less than (<), equal to (=), greater than or equal to (.gtreq.), and less than or equal to (.ltreq.). Value 490 provides the specific quantities that set the constraints for the corresponding attributes. The quantities may be numbers with some known meaning or symbolic terms. For the example shown in FIG. 4B, there are two constraints: the attribute EValue 472 has the Operator Greater Than 482 and a Value of 50 (Item 492), the attribute File Date 474 has the Operator Less Than 484 and a Value of TODAY-2 (Item 494). To change the value, the user enter the desired value in the appropriate spaces 492 or 494. If the user is satisfied with the constraints, pressing the Icon OK 462 ends the session and the constraints are saved. Pressing the Icon Cancel

• 464 retains the original constraints.

DEPR:

Referring to FIG. 5, an example showing the processing of an input document from a parent SmartFolder to folders at successive levels. A Folder Tree 500 consists of three levels. The input document 510 may be any one of a news story, a Word document, an e-mail message, a Power Point presentation, and any other computer readable documents. The root level has the Parent Folder 520. The Input Document 510 is processed to produce relevant attributes that can be compared against the InfoEvaluator Profile 525, and the Constraint Expressions 527 where "k" denotes the tree level (k=0 corresponds the root node). The next level consists of Recent Stories Folder 530, Interesting Stuff-1 Folder 540, and SmartFolder-1 550. In this example, Input Document 510 satisfies the profiles and constraints of Recent Stories Folder 530 and therefore it is stored in Recent Stories Folder 530. Similarly, Input Document 510 satisfies the profiles and constraints of Interesting Stuff-1 Folder 540 and is stored in Interesting Stuff-1 Folder 540. However, Input Document 510 does not completely satisfy the profiles and constraints of SmartFolder-1 550 and therefore it is not stored in SmartFolder-1 550. The last level includes Interesting Stuff-2 Folder 532 and SmartFolder-2 534. The Input Document 510 satisfies the profiles and constraints of Interesting Stuff-2 532 and is stored in Interesting Stuff-2 532. However, Input Document 510 does not satisfy completely the profiles and constraints of SmartFolder-2 534 and is not stored in SmartFolder-2 534.

DEPR:

Referring to FIG. 6, a structure diagram illustrating the operation of one embodiment of the present invention is shown. SmartFolder Structure 600 is the overall organizational structure of the SmartFolder which interfaces to the operating system Windows.RTM. 95 via Shell Interface 680. As is understood by persons of ordinary skill in the art, the use of Windows.RTM. 95 is for illustrative purpose. Any suitable operating system or platform can be utilized. SmartFolder Server 610 represents the environment over which the folders in SmartFolder are created and maintained. The environment may be a single workstation, a desktop computer, a notebook computer, or a computer in a network. SmartFolder Server 610 includes InfoEvaluator Loader 620, Program Interface 640, and Shell Interface 680.

DEPR:

InfoEvaluator Loader 620 loads the evaluation algorithm specified for this folder hierarchy. Program Interface 640 provides the programming interface to the SmartFolder system. Shell Interface 680 provides the interface to the operating system users' interface in operating system shell 682.

DEPR:

The Submit Process 652 submits a document to be categorized by the folder and its children. The Reject Process 654 trains the system to reject a mis-categorized document. The InfoObject instances from Program Interface 640. Submit Process 652, and Reject Process 654 are fed to the Accept Process 660. The Accept Process 660 accepts or rejects a document based on the constraint expressions which may include evaluation values from the InfoEvaluators 670. Expression Analyzer 662 analyzes the constraint expressions and converted them into appropriate format for processing. InfoEvaluators 670 obtains the InfoObject instances from Expression Analyzer 662 and returns the Evaluate. On the basis of the Evaluate(s) returned by InfoEvaluators 670 and other constraint expressions, the Accept Process 660 sends the result of whether a document is accepted or rejected in a folder back to the Submit Process 652 and Program Interface 640.

DEPR:

The process S700 begins at the START state. From the START state, the process S700 determines if new folders are to be created (Step S710). A folder here refers to any storage element that can be organized as part of a filing system. If a new folder is to be created, process S700 proceeds to step S715; otherwise, process S700 goes to a decision step S730. In step S715, the user enters the Categorize dialog interactive session. The user creates the Smartfolders and arrange the hierarchy of the folders according to his or her preference (Step S720). Next, the user generates the initial profile file and constraint file for each of the created Smartfolders (Step S725). Process S700 then returns back to the decision step S710. In decision step S730, it is determined if there is any new information or document to be filed. The arrival of a new document to be filed is normally flagged by the reporter. If there is no new document to be filed, process S700 returns to decision step S710. If there is new document to be filed, process S700 proceeds to step S735. In step S735, the features of the new document are extracted and forwarded to the Constraint Evaluator for determining what folder the document is filed into.

DEPR:

The filing is carried out by going through all the created folders. The folder index k is initialized in step S738. The Constraint Evaluator then obtains the profile and constraint files of the Smartfolder k (Step S750). Using the InfoEvaluator and other

rules, the Constraint Evaluator evaluates the document expressions (Step S755). The document expressions may include the information values of the document and other values such as the document age, size, source, etc. The InfoEvaluator computes the information value for the document based on the profiles of the folder. Other rules to evaluate the document expressions include file data such as time, date, size, source etc. The document expressions contain expressions that correspond to the constraint expression of Smartfolder k. Process S700 then proceeds to decision step S760. In decision step S760, it is determined if the document expressions match the constraint expressions of the Smartfolder k. This step is carried out by comparing the information values and other document values to the corresponding values in the constraint expressions. The combinations of these constraints is also evaluated. These combinations may include logic relations such as AND, OR. The result of the matching is used to determine if the document is to be filed in Smartfolder k.

DEPR:

If the evaluated document expressions match the constraint expressions of Smartfolder k, process S700 proceeds to store the document in Smartfolder k (Step S765). Otherwise, Process S700 skips the storing and moves to the next folder (Step S770). The process S700 determines if another Smartfolder needs to be considered (step S780). If it is, Process S700 returns to step S750. If not, process S700 terminates.

DEPR:

Referring to FIG. 8, a flowchart illustrating a training process S800 for the Submit process is shown. In the Submit process, a folder is trained to accept the document and adjusts the information evaluator as necessary. The document is referred to as an InfoObject. The process begins at the START step. Then the process evaluates the SmartFolder constraints (Step S810). The evaluation utilizes the SmartFolder constraint file (SmartFolder.INI) and the InfoEvaluator information from the server. It is then determined if the InfoObject instances match the constraints at the decision step S820. If NO, the process S800 feeds back an adjustment indicator to the InfoEvaluator Server for this InfoObject (Step S825). The adjustment indicator informs the Information Evaluator to adjust its parameters such that the information value is computed in the direction to satisfy the constraints. If YES, it is next determined if the SmartFolder has sub-folders at the decision step S830. If the SmartFolder does not have any sub-folders, the process goes to step S865 in which the object is stored in the folder and the process is completed. If the SmartFolder has sub-folders, process S800 proceeds with submitting the InfoObject to each sub-folder (Step S835). Each sub-folder then essentially performs similar constraint matching steps to determine if it can accept the InfoObject (Step S840). If it does not accept the InfoObject, process S800 proceeds to decision step S850. If the folder accepts the document, process S800 proceeds to the decision step S860. In the decision step S850, it is determined if there is any constraint that specifies keeping the rejected document. If YES, process S800 proceeds to store the rejected document in the folder (Step S865). If NO, process S800 terminates. In the decision step S860, it is determined if there is any constraint that specifies keeping the accepted document. If NO, process S800 terminates. If YES, process S800 proceeds to store the document in the folder (Step S865).

DEPR:

Referring to FIG. 9, a flowchart illustrating a training process S900 for the Reject process is shown. The Reject process is trained to reject the document and adjusts the Information Evaluator accordingly. From the START state, the process S900 evaluates the SmartFolder constraints (Step S910). The evaluation utilizes the constraint file (SmartFolder. INI) and the InfoEvaluator information from the server. Then process S900 proceeds to the decision step S920 to determine if the document matches all the constraints. If YES, process S900 proceeds to step S925. If NOT, process S900 proceeds to the decision step S930. In step S925, process S900 feeds back an adjustment indicator to the InfoEvaluator server for the document. The adjustment indicator informs the Information Evaluator to adjust its parameters such that the information value is computed in the direction to satisfy the constraints. In the decision step S930, it is determined if the folder is a sub-folder of a SmartFolder. If YES, process S900 proceeds to step S935. If NO, process S900 proceeds to step S940. In step S935, the document is rejected from the parent folder. Then process S900 proceeds to step S940. In step S940 the document is removed from the folder.

DEPR:

The SmartFolder Intelligent Filing System described in the present invention automatically organizes documents based on their contents and the usage profiles. The system eliminates the tedious process of document searching. In addition, the system also learns the usage profile of the user to categorize the documents according to the user's current interests or organizational methods.

L24: Entry 24 of 104

File: USPT

May 4, 1999

US-PAT-NO: 5899995

DOCUMENT-IDENTIFIER: US 5899995 A

TITLE: Method and apparatus for automatically organizing information

DATE-ISSUED: May 4, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Millier; Marshall A.	Hillsboro	OR	N/A	N/A
Pandit; Milind S.	Beaverton	OR	N/A	N/A
Miller; John David	Beaverton	OR	N/A	N/A

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE CODE
Intel Corporation	Santa Clara	CA	N/A	N/A	02

APPL-NO: 8/ 884755

DATE FILED: June 30, 1997

INT-CL: [6] G06F 17/30

US-CL-ISSUED: 707/102; 707/104, 707/6

US-CL-CURRENT: 707/102; 707/104.1, 707/6

FIELD-OF-SEARCH: 707/2, 707/6, 707/10, 707/102, 707/104, 707/501

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

		Search Selected	Search All	
PAT-NO	ISSUE-DATE	PATENTEE-NAME		US-CL
<input type="checkbox"/> 5717914	February 1998	Husick et al.		707/5
<input type="checkbox"/> 5754939	May 1998	Herz et al.		455/4.2
<input type="checkbox"/> 5761662	June 1998	Dasan		707/10
<input type="checkbox"/> 5784608	July 1998	Meske et al.		707/2

ART-UNIT: 271

PRIMARY-EXAMINER: Black; Thomas G.

ASSISTANT-EXAMINER: Loomis; John C.

ATTY-AGENT-FIRM: Blakely, Sokoloff, Taylor & Zafman

ABSTRACT:

The present invention discloses an electronic filing system for automatically organizing information into a number of storage elements or folders. The storage elements are folders arranged hierarchically. The information is stored into the storage elements based on its contents. Each storage element is associated by a set of profiles and a set of constraint expressions. The constraint expressions are quantitative measurements of the profiles. The information is scanned and its features are extracted and compared with the profiles. If the features match the storage element's profiles, the features are quantitatively evaluated to produce information expressions. If the information expressions are within the constraints specified by the constraint expressions, the information is stored into the storage element. The profiles and the constraint

- information is stored into the storage element. The profiles and the constraint expressions can be adjusted to reflect the changes in the user's interests.

15 Claims, 11 Drawing figures

L6: Entry 25 of 55

File: USPT

Jan 12, 1999

US-PAT-NO: 5860066

DOCUMENT-IDENTIFIER: US 5860066 A

TITLE: Imaging and workflow system

DATE-ISSUED: January 12, 1999

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Rouse; Lonnie D.	Largo	FL	N/A	N/A

ASSIGNEE-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY	TYPE	CODE
Payment Systems for Credit Unions Inc.	St. Petersburg	FL	N/A	N/A	02	

APPL-NO: 8/ 671220

DATE FILED: June 27, 1996

INT-CL: [6] G06F 17/40

US-CL-ISSUED: 705/1; 707/500

US-CL-CURRENT: 705/1; 707/500

FIELD-OF-SEARCH: 707/500, 707/7, 364/478.01, 364/478.02, 364/478.03, 705/1-2, 705/7, 705/9, 706/45, 345/326

PRIOR-ART-DISCLOSED:

U.S. PATENT DOCUMENTS

		Search Selected	Search All	
PAT-NO	ISSUE-DATE	PATENTEE-NAME		US-CL
<input type="checkbox"/> <u>5182705</u>	January 1993	Barr et al.		705/11
<input type="checkbox"/> <u>5307262</u>	April 1994	Entel		705/2
<input type="checkbox"/> <u>5319543</u>	June 1994	Cuilhilm		705/3
<input type="checkbox"/> <u>5344132</u>	September 1994	LeBrun et al.		271/35
<input type="checkbox"/> <u>5557515</u>	September 1996	Abbruzzese et al.		705/7
<input type="checkbox"/> <u>5602936</u>	February 1997	Green et al.		382/140
<input type="checkbox"/> <u>5625465</u>	April 1997	Lech et al.		358/448
<input type="checkbox"/> <u>5659746</u>	August 1997	Bankent et al.		707/205
<input type="checkbox"/> <u>5666490</u>	September 1997	Gillings et al.		395/200.68

OTHER PUBLICATIONS

Visual WorkFlo-13 Perspectives, FileNet Corporation, 1995.

FileNet Visual WorkFlo Software--Intelligent Work Management, FileNet Corporation, 1994.

ART-UNIT: 276

PRIMARY-EXAMINER: Feild; Joseph H.

ATTY-AGENT-FIRM: Roberts & Brownell LLC

ABSTRACT:

An imaging and workflow method, and a system for processing information efficiently for a service industry, the system being particularly well suited for use in the cardholder service industry. The system includes support for document scanning, automated rules-based workcase processing, statistical reporting, document generation and document storage and retrieval. This method and system takes advantage of imaging technology to assist the user in scanning information into the system and software modules to improve the processing of workcases. The system also includes a database table that identifies to application processing logic the types and sequences of actions to be implemented for defined workcase types. The computer program utilized by the system is modularized in order to facilitate adding new functions into the system. These new functions are supported through entries in the database tables.

16 Claims, 7 Drawing figures

WEST

 Generate Collection

L6: Entry 33 of 55

File: USPT

May 27, 1997

US-PAT-NO: 5634129

DOCUMENT-IDENTIFIER: US 5634129 A

TITLE: Object oriented system for representing physical locations

DATE-ISSUED: May 27, 1997

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Dickinson; Robert D.	Hayward	CA	N/A	N/A

US-CL-CURRENT: 709/315